EVM User's Guide: LP5810 LP5811 LP5812 LP5813

LP5813 Sample Code

Description

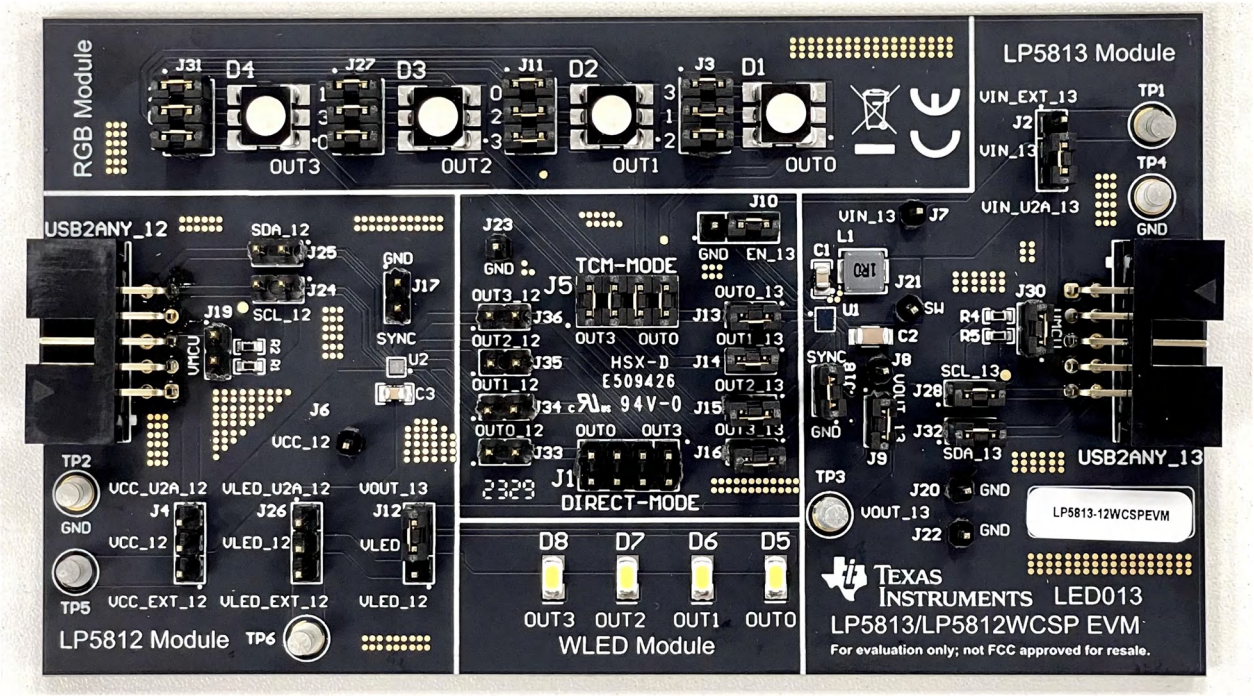
This document describes the preparation and usage of the sample code for the LP5813 device when paired with a LP-MSPM0L1306. Following the instructions provided for setup, the installed code lights up the LEDs on the LP5813-12WCSPEVM and LP5813-12WSONEVM. The LP5813 device can use a standard I2C module of the MCU to communicate to the LED drivers.

Features

- LP5813 Synchronous Boost 4 × 3 Matrix RGB LED driver with autonomous control
- LP5812 4 × 3 Matrix RGB LED driver with autonomous control
- LP5811 synchronous boost 4 channels RGB LED driver with autonomous control
- LP5810 4 channels RGB LED driver with autonomous control
- LP-MSPM0L1306, Arm 32-bit Cortex-M0+, up to 32MHz

Applications

- Portable and wearable electronics: e-cigarette, earbuds and charging case
- Gaming and home entertainment: smart speaker, RGB mouse, video doorbell
- Industrial HMI: EV charger, factory automation



1 Introduction

The sample code showcases the ability to light up the LEDs on the LP5813-12WCSPEVM & LP5813-12WSONEVM. The sample code can suit for all EVM. This helps the user to be able to light up the EVM without any modification to the sample code.

There are four modes in the code: 1) LED_Manual mode; 2) RGB_Manual mode; 3) LED_Auto mode; 4) RGB_Auto mode. The RGB_Manual mode is selected by default. Section 3.2 describes how to work with EVM jumper setting between the modes. In the **LED_Manual** mode, MCU can control 4 white LEDs on the EVM to realize animation patter by sending the I2C commands in real time; In the **RGB_Manual** mode, MCU can control 4 RGB LEDs on the EVM to realize animation patter by sending the I2C commands in real time; In the **LED_Auto** mode, MCU can config the EVM by sending one-time commands, the LED driver can control the 4 white LEDs to play the pre-defined patter using internal animation engine by itself. In the **RGB_Auto** mode, MCU can config the EVM by sending one-time commands, the LED driver can control the 4 RGB LEDs to play the pre-defined patter using internal animation engine by itself.

More detail about the system specification can be found in Section 3.2.

2 Software Setup

To set up the software for the MSPM0L1306 LaunchPad™, please follow the below steps:

1. Download and install Code Composer Studio™.
2. Download and install MSPM0-SDK.
3. Setup the environment.
4. Download and import sample code, Importing the Code Composer Studio (CCS) project according to the process provided in the link: Importing a CCS Project.
5. Load the program according to the process provided in the link: Building and Running Your Project.

2.1 CCS Installation

This section details setps and tips for CCS installation. Remember to save CCS at the address and the default installation place that is suggested.

1. Download Code Composer Studio integrated development environment (IDE) (Version 12.8.0), start installation and keep pressing *Next*.

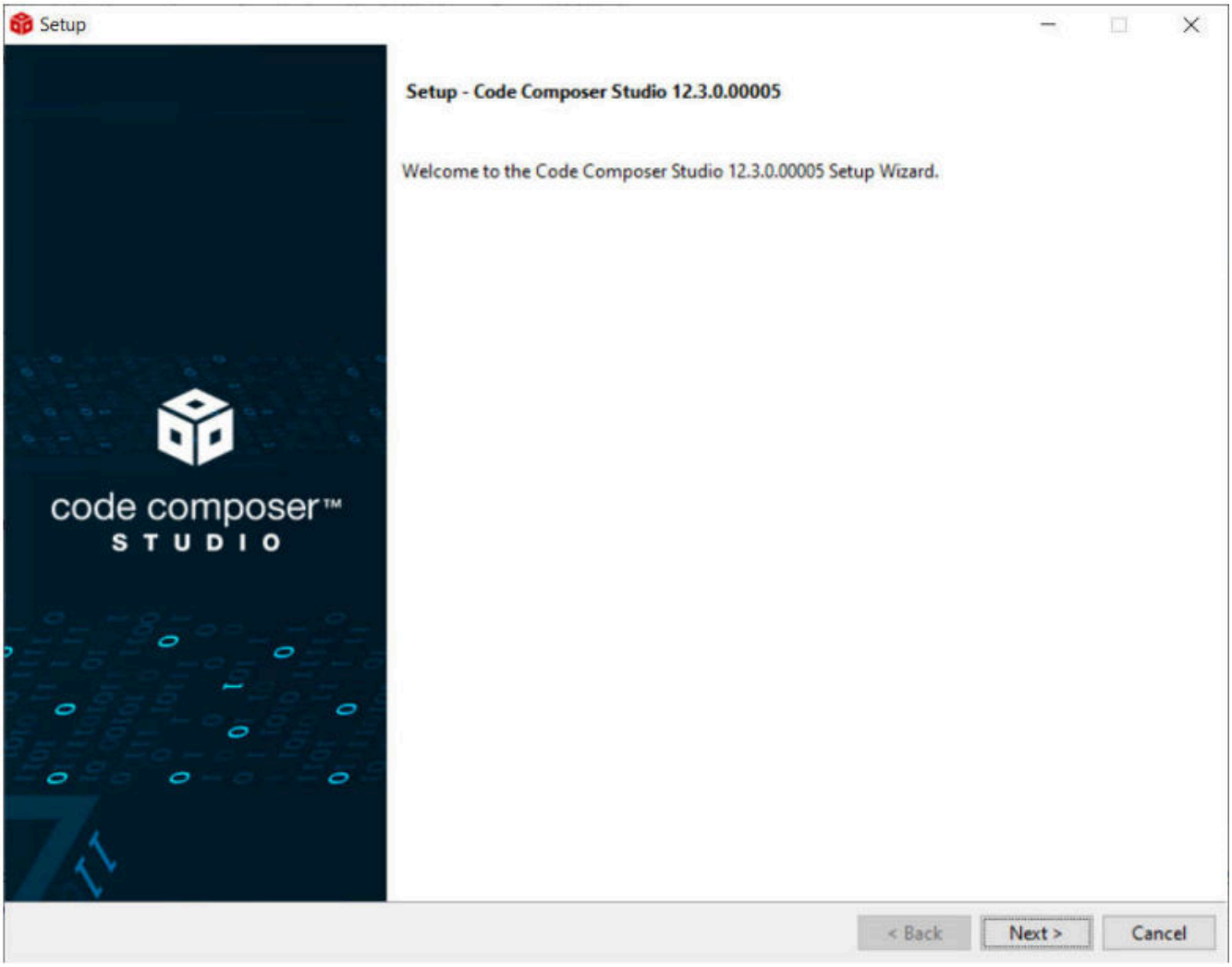


Figure 2-1. CCS Installation

2. Select MSPM0 support component.

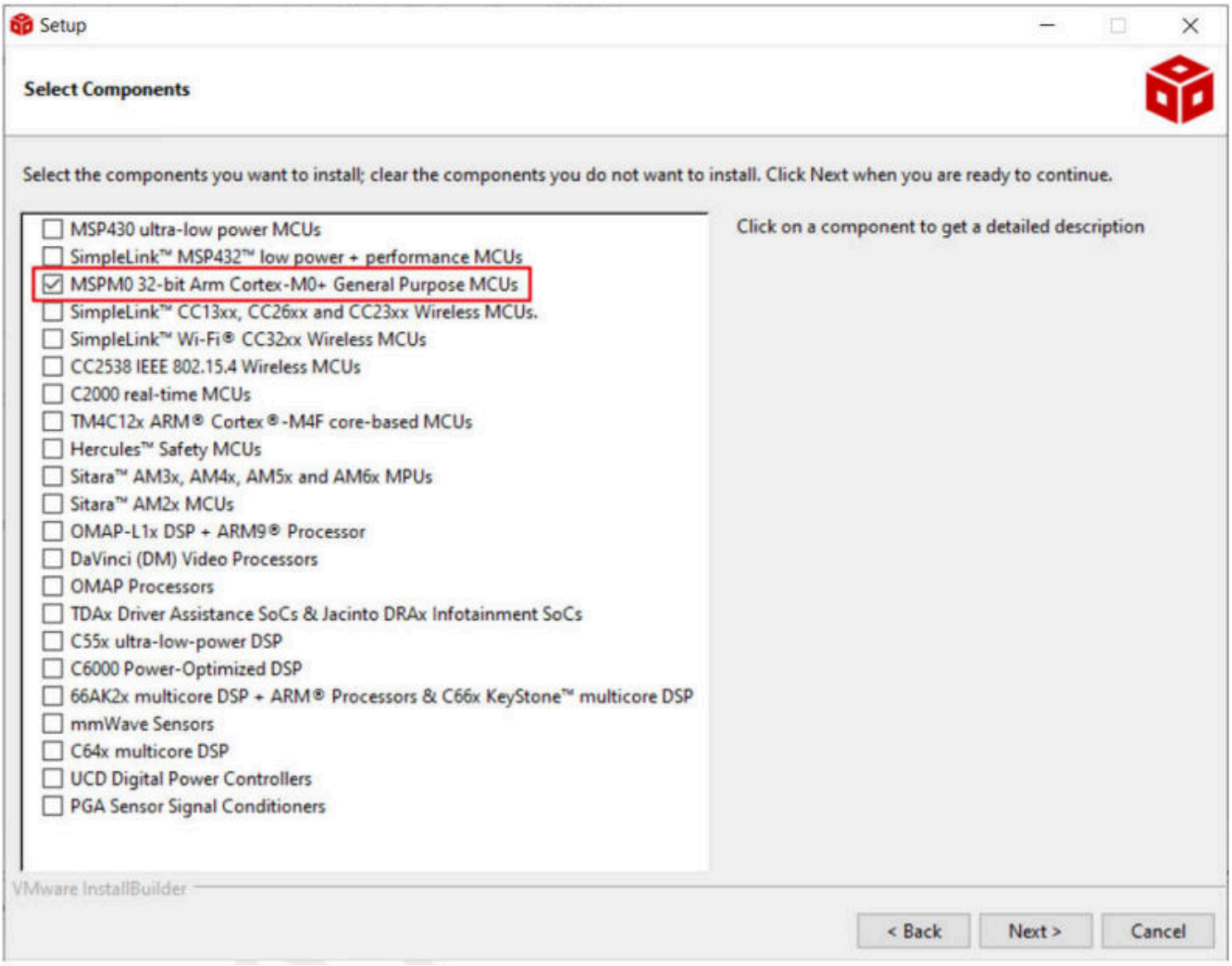


Figure 2-2. MSPM0 Support Selection

3. Select J-link if required.

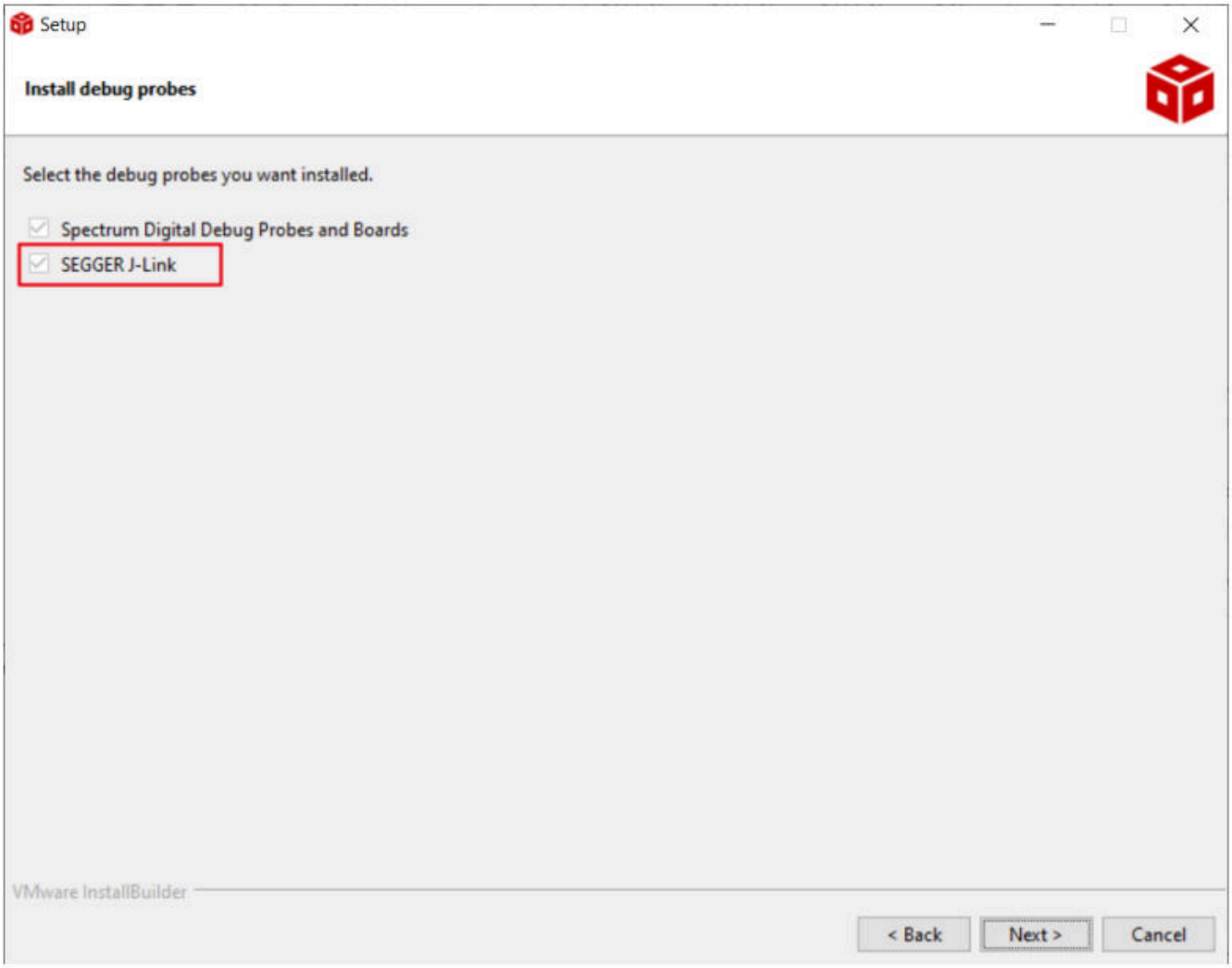


Figure 2-3. J-Link Selection

2.2 MSPM0-SDK Installation

The MSPM0-SDK provides the ultimate collection of software, tools, and documentation to accelerate the development of applications for the MSPM0 MCU platform, providing a consistent and cohesive experience with a wide variety of drivers, libraries, and examples under a single software package. This section details steps to install MSPM0-SDK. After installation, the default SDK directory path is: *C:\ti\mspm0_sdk_x_xx_xx_xx*.

1. Before downloading, a myTI account is required. Register for a myTI account here.
2. Download the latest MSPM0-SDK from the product page. Click Download options, select the operating system, and click the file name to start downloading.

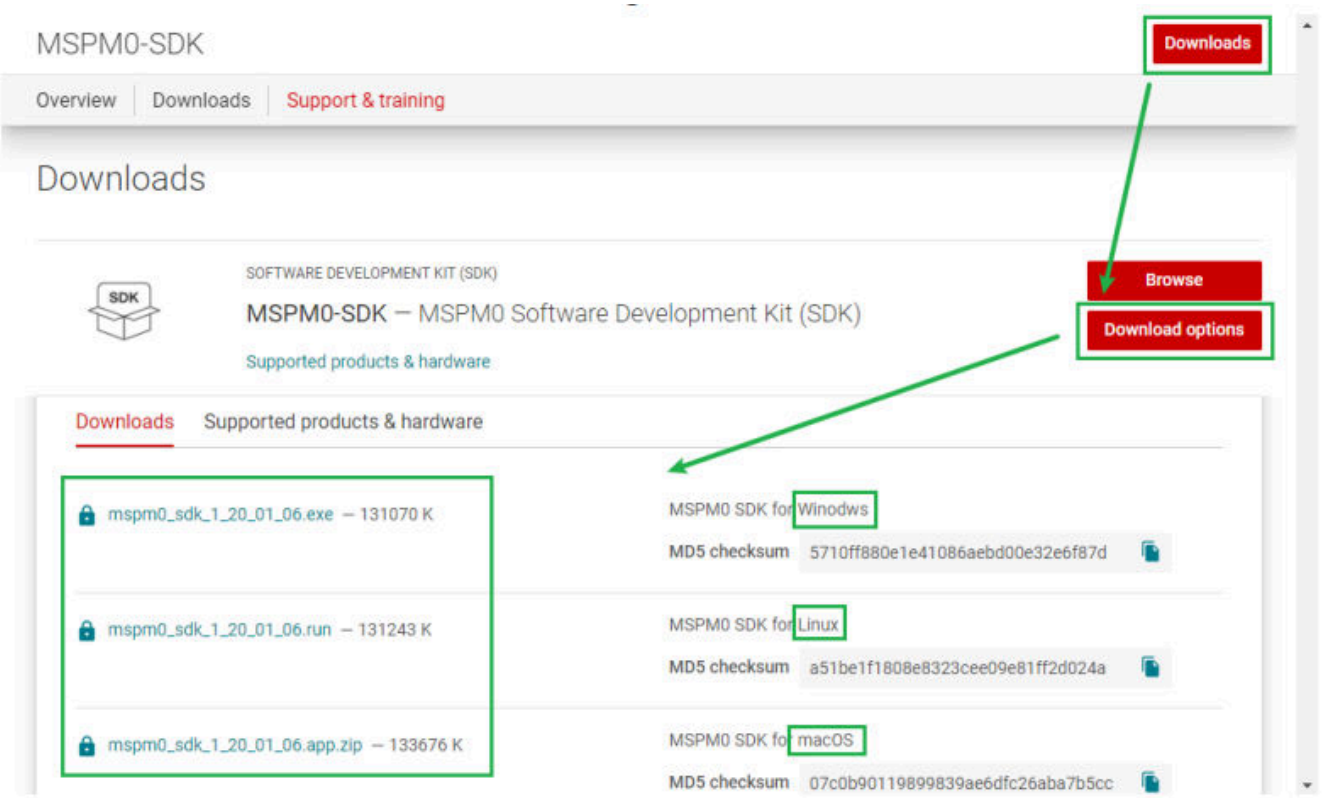


Figure 2-4. MSPM0-SDK Download

3. After downloading, follow the steps in Figure 3-4 to finish installation.

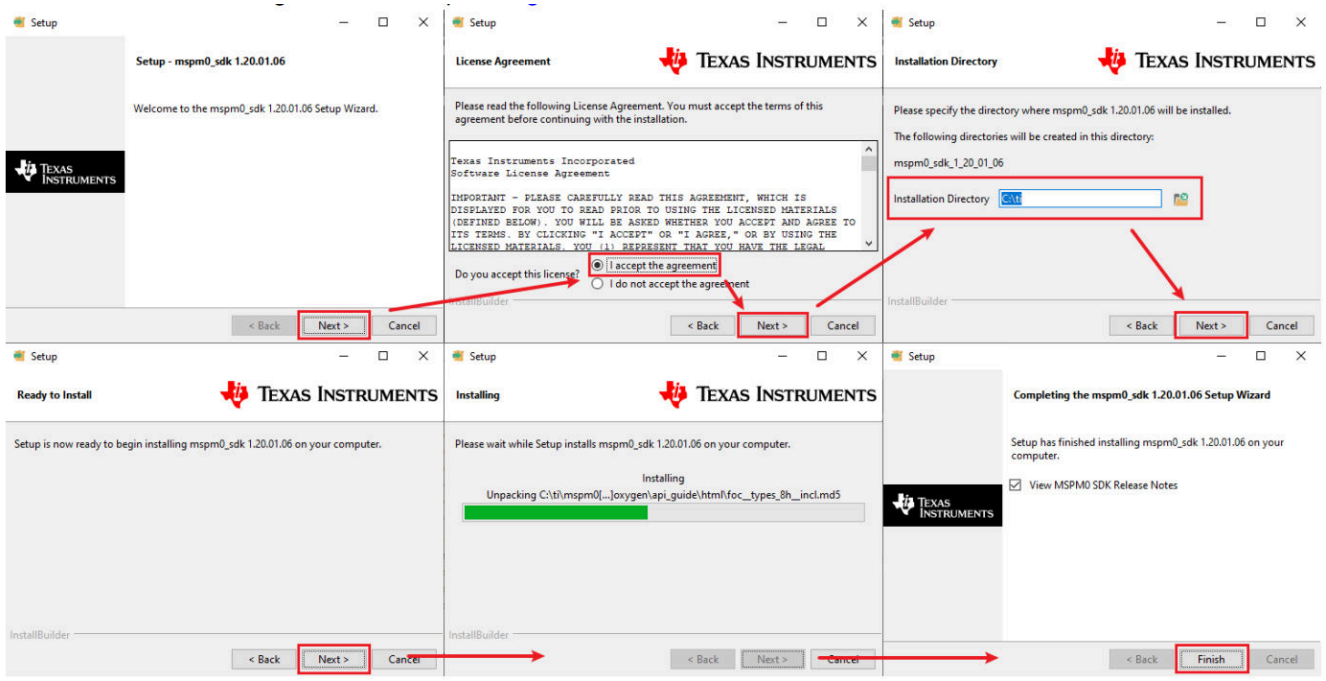


Figure 2-5. MSPM0-SDK Install Step-by-Step

2.3 Environment Setup

For SDK introduction and installation, see Section 3.2.2. If CCS and SDK are installed at the customized address, use the following steps so that CCS loads SDK successfully.

1. Select Window → Preferences.
2. As the MSPM0 SDK is installed in the C:\ drive, add C:\ as the product discovery path.
3. Refresh the Discovered products window. The MSPM0 SDK is recognized automatically.
4. Click the Apply and Close button. The new imported project loads the SDK automatically.

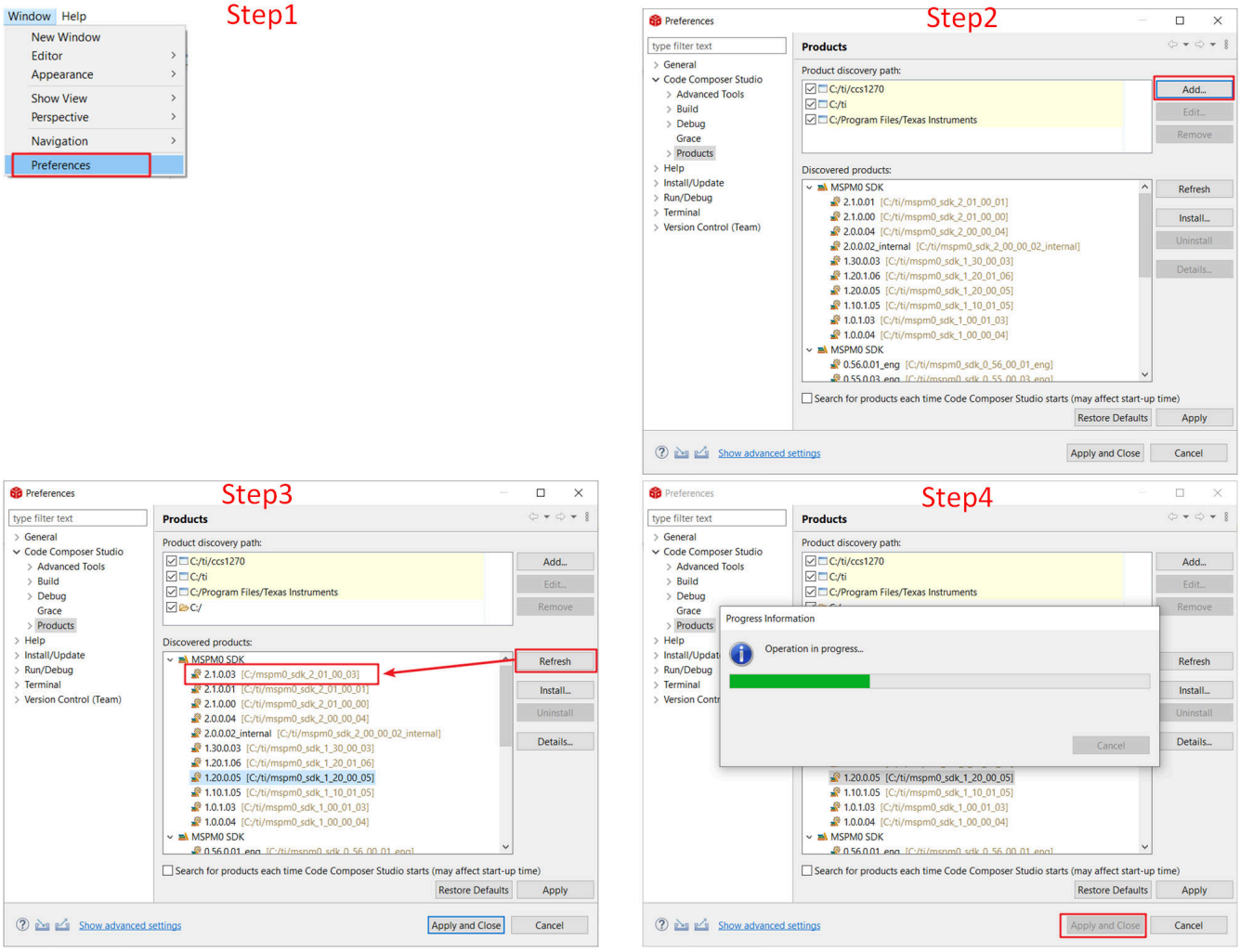


Figure 2-6. Load SDK Product

3 Sample Code Structure

3.1 Flow Diagram

Figure 3-1 depicts the flow in the sample code.

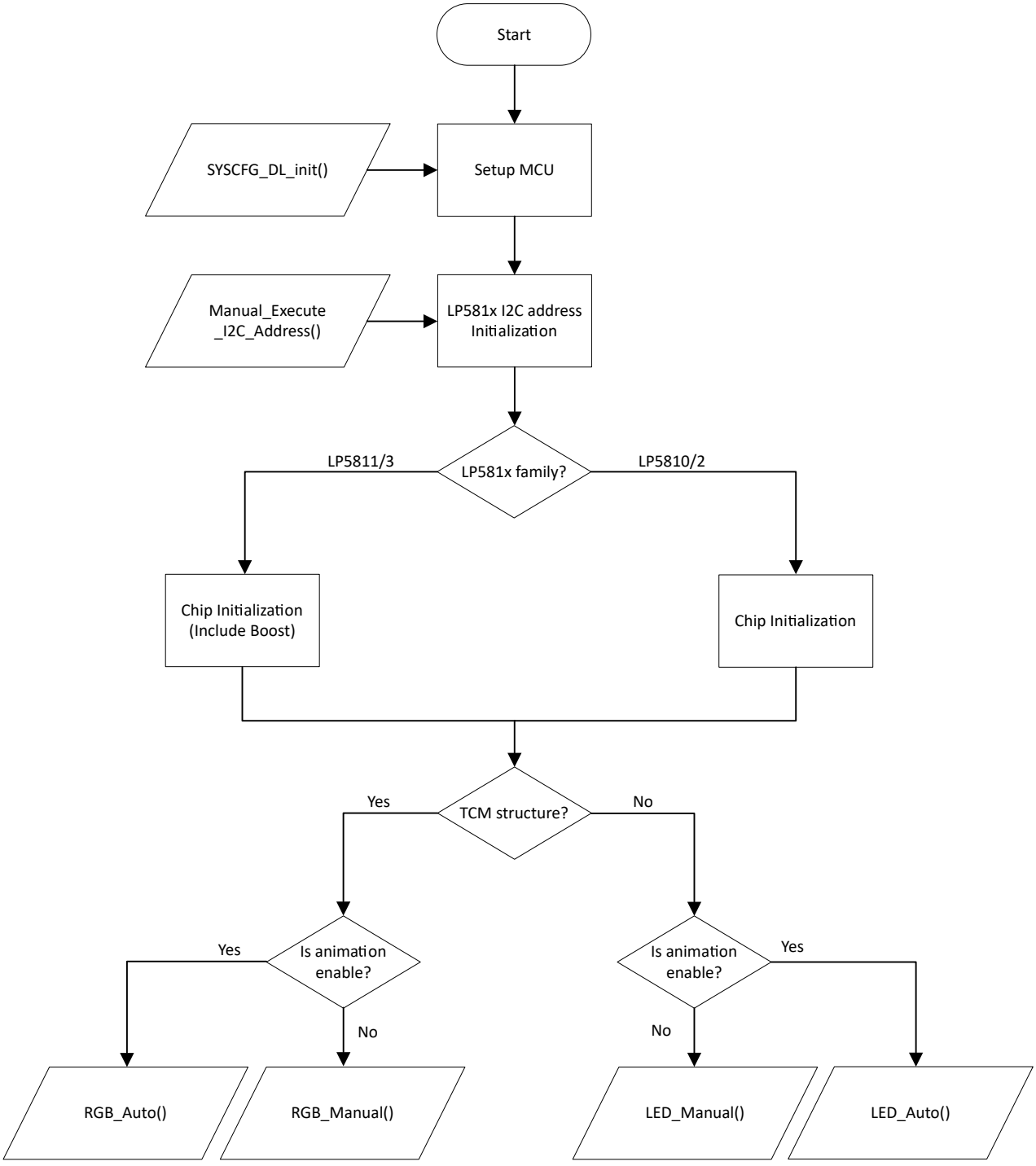


Figure 3-1. Sample Code Flow Diagram

3.2 System Setup

This section describes how the sample code works with the EVM's jumper setting.

The code supports:

- LP5813
- LP5812
- LP5811
- LP5810
- LP5813-12WCSPEVM
- LP5813-12WSONEVM
- LP5811-10EVM

According to the flow diagram, we can see we have 4 different functions to implement different LED effect. **RGB_Auto()** function means that using internal animation function & TCM structure to control 4 RGB LEDs without real time I2C communication. **RGB_Manual()** means that using TCM structure to control 4 RGB LEDs with real time I2C communication to realize LED effect. **LED_Auto()** means that using internal animation to control 4 LEDs without real time I2C communication. **LED_Manual()** means that only using real time I2C communication to control 4 LEDs with real time I2C communication.

Here is the detail jumper setting for each function.

1. `RGB_Auto();`

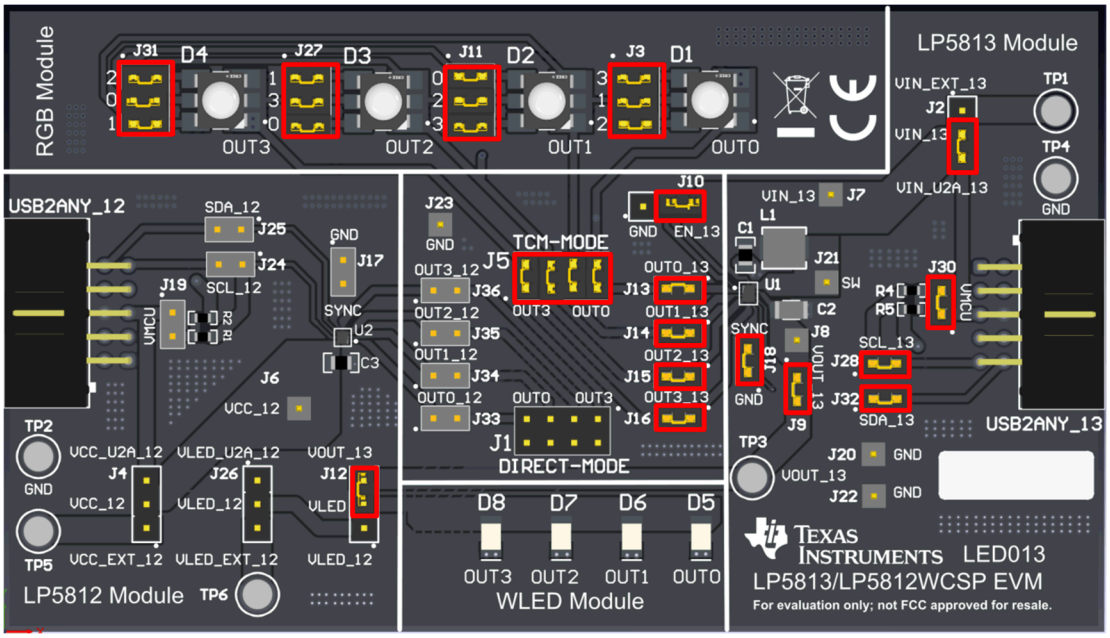


Figure 3-2. 4-Scan Mode and Enable Boost without External Power supply

2. `RGB_Manual();`

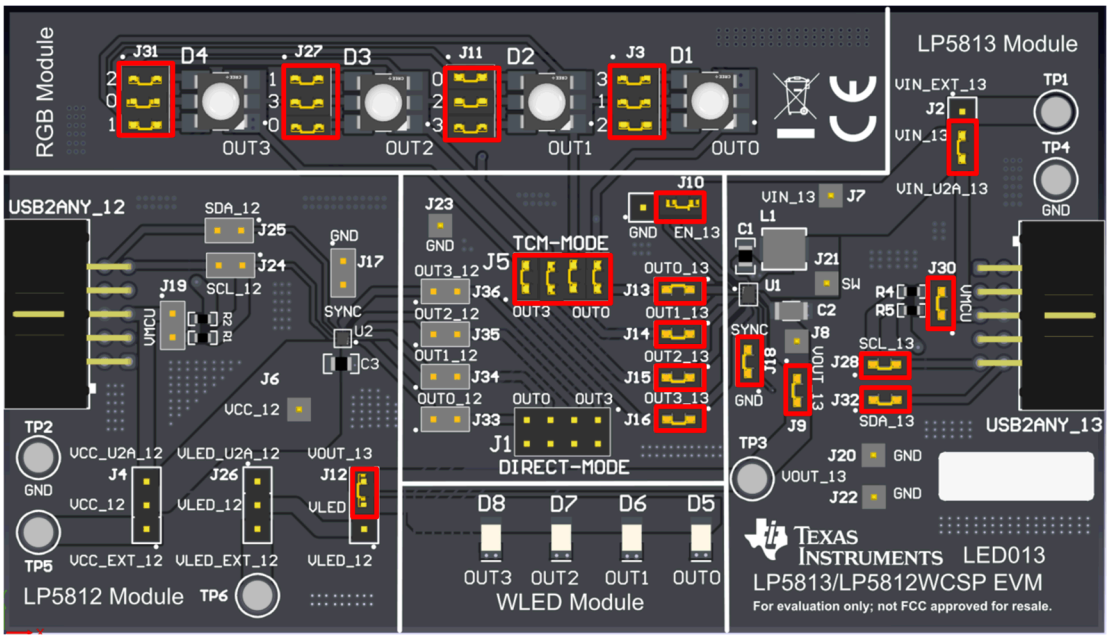


Figure 3-3. 4-Scan Mode and Enable Boost without External Power supply

3.
```
LED_Auto();
```

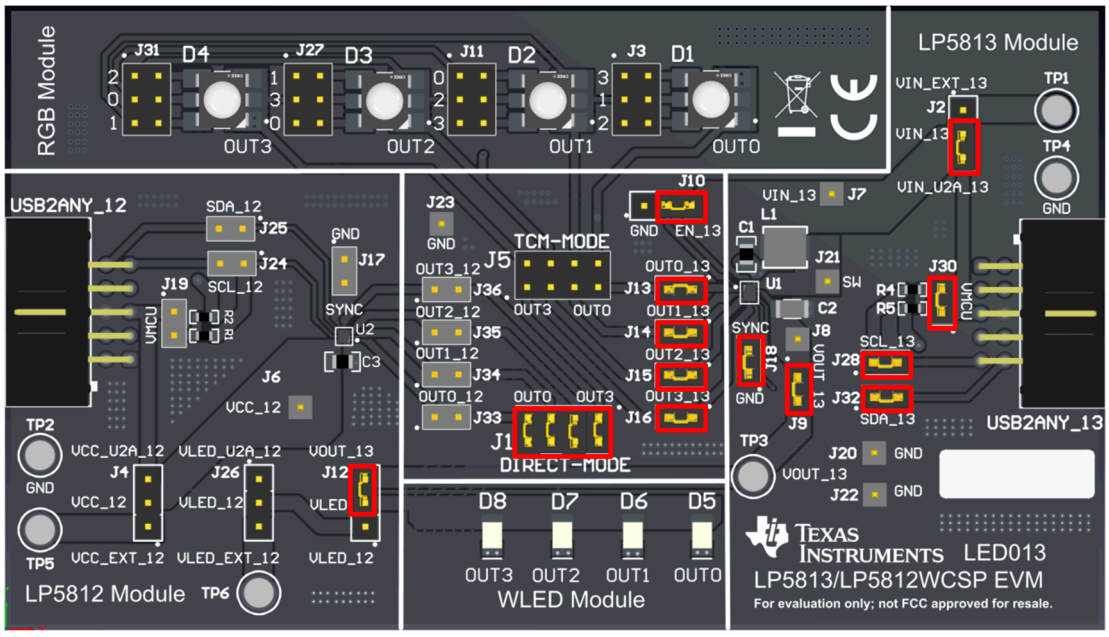


Figure 3-4. Direct mode and Enable Boost without External Power supply

4.
```
LED_Manual();
```

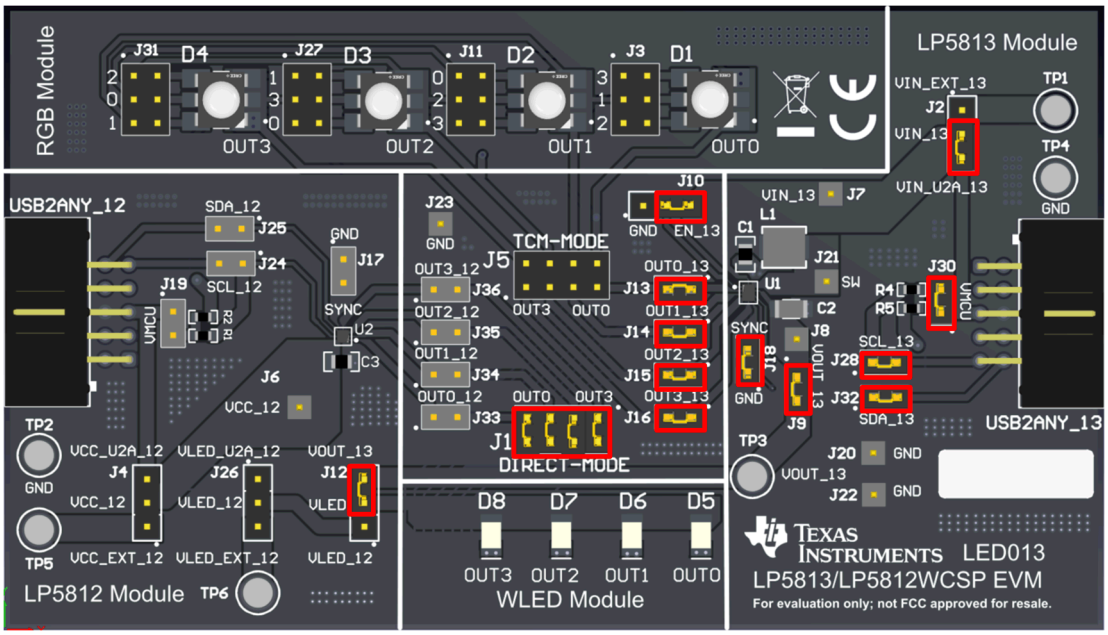


Figure 3-5. Direct mode and Enable Boost without External Power supply

Trademarks

LaunchPad™ and Code Composer Studio™ are trademarks of Texas Instruments.
All trademarks are the property of their respective owners.

IMPORTANT NOTICE AND DISCLAIMER

TI PROVIDES TECHNICAL AND RELIABILITY DATA (INCLUDING DATA SHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES “AS IS” AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS AND IMPLIED, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT OF THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for skilled developers designing with TI products. You are solely responsible for (1) selecting the appropriate TI products for your application, (2) designing, validating and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, regulatory or other requirements.

These resources are subject to change without notice. TI grants you permission to use these resources only for development of an application that uses the TI products described in the resource. Other reproduction and display of these resources is prohibited. No license is granted to any other TI intellectual property right or to any third party intellectual property right. TI disclaims responsibility for, and you will fully indemnify TI and its representatives against, any claims, damages, costs, losses, and liabilities arising out of your use of these resources.

TI’s products are provided subject to TI’s Terms of Sale or other applicable terms available either on ti.com or provided in conjunction with such TI products. TI’s provision of these resources does not expand or otherwise alter TI’s applicable warranties or warranty disclaimers for TI products.

TI objects to and rejects any additional or different terms you may have proposed.

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265